

BBBBBBBBBBBBBBB AAAAAAAA SSSSSSSSSSSS RRRRRRRRRRRRR TTTTTTTTTTTTTTTT LLL
BBBBBBBBBBBBBBB AAAAAAAA SSSSSSSSSSSS RRRRRRRRRRRRR TTTTTTTTTTTTTTTT LLL
BBBBBBBBBBBBBBB AAAAAAAA SSSSSSSSSSSS RRRRRRRRRRRRR TTTTTTTTTTTTTTTT LLL
BBB BBB AAA AAA SSS
BBBBBBBBBBBBBBB AAA AAA SSSSSSSSS
BBBBBBBBBBBBBBB AAA AAA SSSSSSSSS
BBBBBBBBBBBBBBB AAA AAA SSSSSSSSS
BBB BBB AAAAAAAA SSS
BBB BBB AAAAAAAA SSS
BBB BBB AAAAAAAA SSS
BBB BBB AAA AAA SSS
BBBBBBBBBBBBBBB AAA AAA SSSSSSSSSSSS
BBBBBBBBBBBBBBB AAA AAA SSSSSSSSSSSS
BBBBBBBBBBBBBBB AAA AAA SSSSSSSSSSSS

FILEID**BASGETRFA

| | | | | | | | | | | | | |
|----------|--------|--------|----------|-----------|-----------|----------|----------|---------|----|----|---------|----|
| BBBBBBBB | AAAAAA | SSSSSS | GGGGGGGG | EEEEEEEEE | TTTTTTTTT | RRRRRRRR | FFFFFFF | AAAAAA | | | | |
| BBBBBBBB | AAAAAA | SSSSSS | GGGGGGGG | EEEEEEEEE | TTTTTTTTT | RRRRRRRR | FFFFFFF | AAAAAA | | | | |
| BB | BB | AA | AA | SS | GG | EE | TT | RR | RR | FF | AA | AA |
| BB | BB | AA | AA | SS | GG | EE | TT | RR | RR | FF | AA | AA |
| BB | BB | AA | AA | SS | GG | EE | TT | RR | RR | FF | AA | AA |
| BB | BB | AA | AA | SS | GG | EE | TT | RR | RR | FF | AA | AA |
| BBBBBBBB | AA | AA | SSSSSS | GG | EEEEEEE | TT | RRRRRRRR | FFFFFFF | AA | AA | | |
| BBBBBBBB | AA | AA | SSSSSS | GG | EEEEEEE | TT | RRRRRRRR | FFFFFFF | AA | AA | | |
| BB | BB | AAAAAA | SS | GG | GGGGGG | EE | TT | RR | RR | FF | AAAAAAA | |
| BB | BB | AAAAAA | SS | GG | GGGGGG | EE | TT | RR | RR | FF | AAAAAAA | |
| BB | BB | AA | AA | SS | GG | EE | TT | RR | RR | FF | AA | AA |
| BB | BB | AA | AA | SS | GG | EE | TT | RR | RR | FF | AA | AA |
| BBBBBBBB | AA | AA | SSSSSS | GGGGGG | EEEEEEE | TT | RR | RR | FF | AA | AA | |
| BBBBBBBB | AA | AA | SSSSSS | GGGGGG | EEEEEEE | TT | RR | RR | FF | AA | AA | |

| | | |
|----------|--------|--------|
| LL | IIIIII | SSSSSS |
| LL | IIIIII | SSSSSS |
| LL | II | SS |
| LLLLLLLL | IIIIII | SSSSSS |
| LLLLLLLL | IIIIII | SSSSSS |

```
1 0001 0 MODULE BASSGETRFA (
2 0002 0 IDENT = '1-004'
3 0003 0 ) =
4 0004 1 BEGIN
5
6 0006 1 ****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 * ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 * TRANSFERRED.
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 * CORPORATION.
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 ****
28 0028 1 *
29 0029 1 *
30 0030 1 ++
31 0031 1 FACILITY: Basic Language Support
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1    BASSGETRFA will fetch the RFA stored in the RAB.
36 0036 1
37 0037 1 ENVIRONMENT: Runs at any access mode - AST reentrant
38 0038 1
39 0039 1 AUTHOR: Pamela Levesque, CREATION DATE: 2-Jun-1982
40 0040 1
41 0041 1 MODIFIED BY:
42 0042 1
43 0043 1    1-001 - Original. PLL 02-Jun-1982
44 0044 1    1-002 - Make routine global. PLL 3-Jun-1982
45 0045 1    1-003 - RFA is passed by ref. Also, give an error if there is no rfa.
46 0046 1       PLL 4-Jun-1982
47 0047 1    1-004 - Set up ISBSA_USER FP so the unwind in the error handler works
48 0048 1    properly. KC 12-Jun-1984.
49 0049 1 --
50 0050 1
```

```
52      0051 1 %SBTTL 'Declarations'  
53      0052 1  
54      0053 1 SWITCHES:  
55      0054 1  
56      0055 1  
57      0056 1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);  
58      0057 1  
59      0058 1  
60      0059 1 LINKAGES:  
61      0060 1      NONE  
62      0061 1  
63      0062 1  
64      0063 1 TABLE OF CONTENTS:  
65      0064 1  
66      0065 1  
67      0066 1 FORWARD ROUTINE  
68      0067 1      BASSGETRFA : NOVALUE;                      ! Get RFA from RAB  
69      0068 1  
70      0069 1  
71      0070 1 INCLUDE FILES:  
72      0071 1  
73      0072 1  
74      0073 1 LIBRARY 'RTLSTARLE';                      ! System symbols, typically from SYSLIBRARY:STARLET.L32  
75      0074 1  
76      0075 1 REQUIRE 'RTLIN:RTLPSECT';                  ! Define PSECT declarations macros  
77      0170 1  
78      0171 1 REQUIRE 'RTLML:OTSIISB';  
79      0339 1  
80      0340 1 REQUIRE 'RTLML:OTSLUB';  
81      0480 1  
82      0481 1 REQUIRE 'RTLIN:OTSLNK';  
83      0910 1  
84      0911 1 MACROS:  
85      0912 1  
86      0913 1      NONE  
87      0914 1  
88      0915 1 EQUATED SYMBOLS:  
89      0916 1  
90      0917 1      NONE  
91      0918 1  
92      0919 1 FIELDS:  
93      0920 1  
94      0921 1      NONE  
95      0922 1  
96      0923 1 PSECTS:  
97      0924 1  
98      0925 1 DECLARE_PSECTS (BAS);                      ! Declare PSECTs for BASS$ facility  
99      0926 1  
100     0927 1 OWN STORAGE:  
101     0928 1  
102     0929 1      NONE  
103     0930 1  
104     0931 1 EXTERNAL REFERENCES:  
105     0932 1  
106     0933 1  
107     0934 1 EXTERNAL ROUTINE  
108     0935 1      BASS$STOP_IO : NOVALUE,                      ! Signal fatal I/O error
```

```
: 109      0936 1     BAS$CB_PUSH : JSB_CB_PUSH NOVALUE,      ! Load register CCB
: 110      0937 1     BAS$CB_POP : JSB_CB_POP NOVALUE;      ! Done with register CCB
: 111      0938 1
: 112      0939 1     EXTERNAL LITERAL
: 113      0940 1     BASSK_NO_CURREC : UNSIGNED (8),      ! Condition value symbols
: 114      0941 1     BASSK_ILCILLACC : UNSIGNED (8),      ! No current record
: 115      0942 1     BASSK_IO_CHANOT : UNSIGNED (8);      ! Illegal or illogical access
: 116      0943 1

```

```

118 0944 1 %SBTTL 'BASS$GETRFA - Get RFA from RAB'
119 0945 1 GLOBAL ROUTINE BASS$GETRFA (
120 0946 1     UNIT,
121 0947 1     RFA
122 0948 1 ) : NOVALUE =
123 0949 1
124 0950 1 +++
125 0951 1 | FUNCTIONAL DESCRIPTION:
126 0952 1
127 0953 1 | This routine returns the RFA (record file address) of the last record
128 0954 1 | accessed for the specified channel. The 6 byte RFA is stored in the
129 0955 1 | location passed as a dtype z descriptor.
130 0956 1
131 0957 1 | CALLING SEQUENCE:
132 0958 1
133 0959 1 | BASS$GETRFA (UNIT.rlu.v, RFA.wx.r)
134 0960 1
135 0961 1 | FORMAL PARAMETERS:
136 0962 1
137 0963 1     UNIT.rlu.v      logical unit number
138 0964 1     RFA.wx.r       where to store the RFA
139 0965 1
140 0966 1 | IMPLICIT INPUTS:
141 0967 1
142 0968 1     NONE
143 0969 1
144 0970 1 | IMPLICIT OUTPUTS:
145 0971 1
146 0972 1     NONE
147 0973 1
148 0974 1 | COMPLETION STATUS:
149 0975 1
150 0976 1     Signals any errors
151 0977 1
152 0978 1 | SIDE EFFECTS:
153 0979 1
154 0980 1     NONE
155 0981 1
156 0982 1 | --
157 0983 1
158 0984 2 | BEGIN
159 0985 2
160 0986 2 | GLOBAL REGISTER
161 0987 2     CCB = K_CCB_REG : REF BLOCK [, BYTE];
162 0988 2
163 0989 2 | BUILTIN
164 0990 2     FP;
165 0991 2
166 0992 2 | LOCAL
167 0993 2     FMP : REF BLOCK [, BYTE];
168 0994 2
169 0995 2     FMP = .FP;
170 0996 2
171 0997 2 | + Allocate the LUB/ISB/RAB for this unit if necessary.
172 0998 2
173 0999 2 | - BASS$CB_PUSH (.UNIT, LUB$K_ILUN_MIN);
174 1000 2

```

```

175 1001 2 1+ Load the CCB with the user's FP so that an unwind in the
176 1002 2 1- error handler works properly.
177 1003 2 1+ CCB [ISBSA_USER_FP] = .FMP [SFSL_SAVE_FP];
178 1004 2 1- If the channel is not open, give an error. There's no RFA for
179 1005 2 1+ channel 0.
180 1006 2 1- IF (NOT .CCB [LUB$V_OPENED]) THEN BASS$STOP_IO (BASSK_IO_CHANOT);
181 1007 2 1+ No RFA for virtual files.
182 1008 2 1- IF .CCB [LUB$V_VA_USE] THEN BASS$STOP_IO (BASSK_ILLILLACC);
183 1009 2 1+ Return the RFA.
184 1010 2 1- IF .CCB [RAB$L_RFA0] NEQ 0
185 1011 2 1+ THEN CH$MOVE (6, CCB [RAB$W_RFA], .RFA)
186 1012 2 1- ELSE BEGIN
187 1013 2 1+ BASS$STOP_IO (BASSK_NO_CURREC)
188 1014 2 1- END;
189 1015 2 1+ Pop the CCB off the I/O system.
190 1016 2 1- BASS$CB_POP ();
191 1017 2 1+ Return the RFA.
192 1018 2 1- IF .CCB [RAB$L_RFA0] NEQ 0
193 1019 2 1+ THEN CH$MOVE (6, CCB [RAB$W_RFA], .RFA)
194 1020 2 1- ELSE BEGIN
195 1021 2 1+ BASS$STOP_IO (BASSK_NO_CURREC)
196 1022 2 1- END;
197 1023 2 1+ Return the RFA.
198 1024 2 1- IF .CCB [RAB$L_RFA0] NEQ 0
199 1025 2 1+ THEN CH$MOVE (6, CCB [RAB$W_RFA], .RFA)
200 1026 2 1- ELSE BEGIN
201 1027 2 1+ BASS$STOP_IO (BASSK_NO_CURREC)
202 1028 2 1- END;
203 1029 2 1+ Pop the CCB off the I/O system.
204 1030 2 1- BASS$CB_POP ();
205 1031 2 1+ Return the RFA.
206 1032 1 1- END;

```

! End of routine BASSGETRFA

| | | | |
|--------|-------------------------------|----------------------|-------------------------|
| .TITLE | BASSGETRFA | | |
| .IDENT | \1-004\ | | |
| .EXTRN | BASS\$STOP_IO, BASS\$CB_PUSH | | |
| .EXTRN | BASS\$CB_POP, BASSK_NO_CURREC | | |
| .EXTRN | BASSK_I[ILLILLACC | | |
| .EXTRN | BASSK_IO_CHANOT | | |
| .PSECT | _BASS\$CODE,NOWRT, SHR, PIC,2 | | |
| | | | : 0945 |
| | | | : 0995 |
| | | | : 0999 |
| | | | : 1005 |
| | | | : 1010 |
| | | | : 1015 |
| | | | : |
| 56 | 0000000G | 00 9E 00000 | |
| 53 | | 5D D0 00009 | |
| 50 | | 08 CE 0000C | |
| 52 | 04 | AC D0 0000F | |
| FF4C | CB | 0000000G 00 16 00013 | |
| | 07 | OC A3 D0 00019 | |
| | 7E | FC AB E8 0001F | |
| | 66 | 00G 8F 9A 00023 | |
| | 07 | 01 FB 00027 | |
| | 7E | FF AB E9 0002A 1\$: | |
| | 66 | 00G 8F 9A 0002E | |
| | | MOVAB | BASS\$STOP_IO, R6 |
| | | MOVL | FP, FMP |
| | | MNEGL | #8, R0 |
| | | MOVL | UNIT, R2 |
| | | JSB | BASS\$CB_PUSH |
| | | MOVL | 12(FMP) - 180(CCB) |
| | | BLBS | -4(CCB), 1\$ |
| | | MOVZBL | #BASSK_IO_CHANOT, -(SP) |
| | | CALLS | #1, BASS\$STOP_IO |
| | | BLBC | -1(CCB), 2\$ |
| | | MOVZBL | #BASSK_ILLILLACC, -(SP) |
| | | CALLS | #1, BASS\$STOP_IO |

BASSGETRFA
1-004

BASSGETRFA - Get RFA from RAB

J 16
16-Sep-1984 00:34:55
14-Sep-1984 11:55:04
VAX-11 Bliss-32 v4.0-742
[BASRTL.SRC]BASGETRFA.B32;1

Page 6
(3)

| | | | | | | | | | |
|----|----|-----------|----|-------------|------|--------|-------------------------|--|------|
| | | 10 | AB | D5 00035 | 2\$: | TSTL | 16(CCB) | | 1020 |
| 08 | BC | 10 | AB | 08 13 00038 | | BEQL | 3\$ | | 1022 |
| | | | | 06 28 0003A | | MOVC3 | #6, 16(CCB), @RFA | | 1025 |
| | | | | 07 11 00040 | | BRB | 4\$ | | 1030 |
| 7E | | 00G | 8F | 9A 00042 | 3\$: | MOVZBL | #BASSK NO CURREC, -(SP) | | 1032 |
| 66 | | | | 01 FB 00046 | | CALLS | #1, BASS\$STOP_10 | | |
| | | 00000000G | 00 | 16 00049 | 4\$: | JSB | BASS\$CB_POP | | |
| | | | | 04 0004F | | RET | | | |

; Routine Size: 80 bytes. Routine Base: _BASSCODE + 0000

: 207 1033 1 !<BLF/PAGE>

BASSGETRFA
1-004

BASSGETRFA - Get RFA from RAB

: 209 1034 1 END
: 210 1035 1
: 211 1036 0 ELUDOM

K 16

16-Sep-1984 00:34:55
14-Sep-1984 11:55:04

VAX-11 Bliss-32 v4.0-742
[BASRTL.SRC]BASGETRFA.B32;1

Page 7
(4)

: ! End of module BASSGETRFA

PSECT SUMMARY

| Name | Bytes | Attributes |
|-------------|-------|---|
| _BASS\$CODE | 80 | NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2) |

Library Statistics

| File | ----- | Symbols | ----- | Pages | Processing |
|-------------------------------------|-------|---------|---------|--------|------------|
| | Total | Loaded | Percent | Mapped | Time |
| \$_255\$DUA2B:[SYSLIB]STARLET.L32;1 | 9776 | 3 | 0 | 581 | 00:01.0 |

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LISS:BASGETRFA/OBJ=OBJ\$:\$BASGETRFA MSRC\$:\$BASGETRFA/UPDATE=(ENH\$:\$BASGETRFA
)

: Size: 80 code + 0 data bytes
: Run Time: 00:08.5
: Elapsed Time: 00:20.3
: Lines/CPU Min: 7338
: Lexemes/CPU-Min: 43912
: Memory Used: 115 pages
: Compilation Complete

0023 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

